Tidal turbine array optimisation using the adjoint approach

S.W. Funke[a,b], P.E. Farrell[a,c], M.D. Piggott[a,b]

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, London, UK*
[b] *Grantham Institute for Climate Change, Imperial College London, London, UK*
[c] *Center for Biomedical Computing, Simula Research Laboratory, Oslo, Norway*

Abstract

Oceanic tides have the potential to yield a vast amount of renewable energy. Tidal stream generation is one of the key technologies for extracting and harnessing this potential. In order to extract an economically useful amount of power, hundreds of tidal turbines must typically be deployed in an array. This naturally leads to the question of where these turbines should be placed within the array area to extract the maximum possible power: the positioning of the turbines could significantly influence the extracted power, and hence is of major economic interest. However, manual optimisation is difficult due to legal site constraints, nonlinear interactions of the turbine wakes, and the cubic dependence of the power on the flow speed. The novel contribution of this paper is the formulation of this problem as an optimisation problem constrained by a physical model, which is then solved using an efficient gradient-based optimisation algorithm. In each optimisation iteration, a two-dimensional finite element shallow water model predicts the flow and the performance of the current array configuration. The gradient of the power extracted with respect to the turbine positions is then computed in a fraction of the time taken for a flow solution by solving the associated adjoint equations. These equations propagate causality backwards through the computation, from the power extracted back to all turbine positions. This yields the gradient at a cost independent of the number of turbines, which is crucial for any practical application. The utility of the approach is demonstrated by optimising turbine arrays in four idealised scenarios and a more realistic case with up to 256 turbines in the Inner Sound of the Pentland Firth, Scotland.

Keywords: marine renewable energy, tidal turbines, gradient-based optimisation, adjoint method, shallow water equations, array layout

Email address: s.funke09@imperial.ac.uk (S.W. Funke)

1. Introduction

With the increasing cost of energy, tidal turbines are becoming a competitive and promising option for renewable electricity generation. A key advantage of tidal energy is that the power extracted is predictable in advance, which is highly attractive for grid management. In order to amortise the fixed costs of installation and grid connection, arrays consisting of hundreds of tidal turbines must typically be deployed at a particular site. This raises the question of where to place the turbines within the site in order to maximise the power output; finding the optimal configuration is of huge importance as it could substantially change the energy captured and possibly determine whether the project is economically viable. However, the determination of the optimal layout is difficult because of the complex flow interactions between turbines and the fact that the power output depends sensitively on the flow velocity at the turbine positions.

This problem has heretofore been addressed in two different ways. One approach is to simplify the tidal flow model such that the solutions are either available as explicit analytical expressions, or are extremely fast to compute. This means that the optimum can be analytically derived, or that the whole parameter space of possible configurations can be rapidly explored. For example, Bryden and Couch (2007) and Garrett and Cummins (2008) optimised simplified models to derive an estimate for the maximum energy that can be extracted from a tidal basin. Vennell (2010, 2011) used simple one-dimensional models to investigate the influence of different farm parameters, such as the drag coefficient used to parameterise the turbines and the number of turbine rows, to deduce optimal configurations. While this approach can provide a coarse estimate for the power potential of a site, these simplified models cannot accurately capture the complex nonlinear flow interactions between turbines.

The second approach is to use more complex flow models to accurately predict the tidal flow, the turbine wakes, and the resulting power output. These models are usually formulated as numerical solutions to partial differential equations (PDEs). The computational expense of these models prohibits the exploration of the whole parameter space (Thomson et al., 2011). Consequently, typically only a handful of manually identified turbine layouts are investigated in a given scenario (Adams et al., 2011). Divett et al. (2013) compared the power output of four different layouts in a rectangular channel by solving the two-dimensional nonlinear shallow water equations and was able to improve the power outcome by over 50% compared to a regular layout. Lee et al. (2010) used a three-dimensional model to investigate how the distance between adjacent rows in a regular array layout impacts the turbine efficiency and showed an efficiency decay for distances of less than three times the turbine diameter. While these studies show the potential of improving the performance by changing the turbine positions, such manual optimisation guided by intuition and experience becomes difficult in a realistic domain with complex bottom bathymetry, flow dynamics and hundreds of turbines.

In this paper, we present a novel technique for maximising the power ex-

traction of array layouts that combines the physical fidelity of PDE-based flow models with advanced automated optimisation techniques that identify the optimal solution in a computationally feasible number of iterations. The turbine layout problem is formulated as a PDE-constrained optimisation problem, which is a major topic of research in applied mathematics (Gunzburger, 2003; Hinze et al., 2009). The resulting maximisation problem is solved using a gradient-based optimisation algorithm that takes orders of magnitude fewer iterations than genetic algorithms or simulated annealing approaches (see e.g. Bilbao and Alba (2009)). In this paper, the power extracted by an array layout is predicted using a two-dimensional nonlinear shallow water model, which captures the interactions between the geometry, the turbines, and the flow. The gradient of the power is efficiently computed using the *adjoint technique* of variational calculus, which solves an auxiliary system that propagates causality backwards through the physical system. This yields the gradient at a cost independent of the number of turbines to be optimised, which is crucial for the method to be applied to large arrays. This gradient is used by the optimisation algorithm to automatically reposition the turbines. The flow solution is re-evaluated, and the algorithm iterated until an optimum is found.

This approach has several key advantages. Firstly, it closes the optimisation loop, by accounting for the effects of the turbines on the flow field itself. This is necessary to find the actual optimum of the nonlinear optimisation problem. Secondly, unlike gradient-free methods, the approach scales to large numbers of turbines, which is necessary for the optimisation of industrial arrays. For example, in section 6, an array of 256 turbines is optimised in a realistic domain at an approximate cost of 200 flow solutions. Thirdly, the optimisation algorithm can incorporate complex constraints such as minimum separation distances, bathymetry gradient constraints, and legal site restrictions. Finally, the same mathematical framework extends naturally to more realistic flow models such as the Reynolds-averaged Navier-Stokes equations, and to other functionals such as profit or environmental impact.

The approach is implemented in an open-source software framework called OpenTidalFarm; all code and examples from this paper are available at `http://opentidalfarm.org`.

1.1. Optimisation algorithms

Optimisation algorithms can be divided into two categories: gradient-free and gradient-based algorithms. Gradient-free optimisation algorithms use the functional of interest (in this case, power extracted by the array) as a black box. They proceed by evaluating the functional at many points in parameter space and use these values to decide which areas merit further exploration. While these methods tend to be robust and can, under certain smoothness conditions, provably find globally optimal solutions (Rudolph, 1996), they typically require a very large number of functional evaluations that scales linearly or superlinearly with the number of parameters to be optimised. For example, Bilbao and Alba (2009) used a genetic algorithm that mimics the process of natural evolution to optimise the location of 8 wind turbines. The algorithm was able to improve

the power output by about 70% compared to the initial layout after $17,300$ functional evaluations. This large number of evaluations clearly introduces a practical upper limit for the number of turbines that can be optimised. This difficulty is compounded if a more realistic (and hence more expensive) model is used.

By contrast, gradient-based optimisation algorithms use additional information to update the position in parameter space at each iteration: the first or higher derivatives of the functional of interest with respect to the parameters. Depending on the problem, this can lead to a significant reduction in the number of iterations required compared to gradient-free algorithms, making these the only feasible choice for large scale optimisation problems (Gunzburger, 2003). One caveat of applying gradient-based optimisation algorithms is that they find only local optima. This issue can be circumvented by using hybrid approaches (Huang, 2009). The main difficulty of applying gradient-based methods is that the gradient computation can be difficult for complex models, as it involves differentiating through the solution of a partial differential equation.

One way to obtain the derivative information is to approximate the gradient using finite differences. However, a major disadvantage of this approach is that a single gradient evaluation requires a large number of functional evaluations that scales linearly with the number of optimisation parameters. This sets a practical upper bound on the number of turbines to be optimised, and discards the main advantage of gradient-based optimisation algorithms. Alternatively, the tangent linearisation of the model can efficiently compute the derivative of all outputs with respect to a single input, while the adjoint linearisation can efficiently compute the derivative of a single output with respect to all inputs. For the turbine optimisation problem, we wish to maximise a single output (the power extracted) with respect to many input parameters (the positions of the turbines); this means that the adjoint approach is the natural choice, as the required gradient information can be computed in a number of equation solves that is independent of the number of turbines.

The development of adjoint models is generally considered as very complicated (Giles and Pierce, 2000; Naumann, 2012). However, this problem has been solved in recent work for the case where the forward model is discretised using finite elements, in the high-level FEniCS framework (Farrell et al., 2012). This allows for the extremely rapid development of optimally efficient adjoint models, which significantly reduces the development effort required to implement gradient-based optimisation algorithms for PDE-constrained optimisation problems (Funke and Farrell, 2013).

To the best of our knowledge, this paper presents the first application of the adjoint method to the optimisation of turbine arrays. While the examples are shown in the marine context, it is expected that the presented techniques can also be applied to the layout optimisation of wind farms. As the wind turbine layout problem is both closely related and better studied, we next review techniques proposed for its solution.

1.2. Wind farm optimisation

Layout optimisation for wind farms has been addressed in numerous studies, most of which are based on gradient-free optimisation algorithms. In particular, evolutionary methods (Bäck, 1996) are known to yield good results (Salcedo-Sanz et al. (2011) and the references therein). These algorithms mimic the process of natural evolution by considering a population of candidate solutions on which it executes an evolutionary process to find the "fittest" solution.

A related method is particle swarm optimisation (Kennedy and Eberhart, 1995), which considers a population of candidate solutions called particles, that move through the parameter space and influence each other to drive the swarm to the best solution. Wan et al. (2010) applied particle swarm optimisation on an analytical wake model to optimise the location of 39 turbines and showed that this approach can yield better optimal solutions than genetic algorithms. Simulated annealing algorithms (Laarhoven and Aarts, 1987) are probabilistic optimisation methods that exploit an analogy between the way in which a metal heats and cools into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more abstract system. Bilbao and Alba (2009) used an analytical wake model to compare a simulated annealing algorithm with a genetic optimisation algorithm. In a scenario with 47 turbines, the number of model evaluations was reduced from $1,036,200$ with the genetic algorithm to $61,802$ with simulated annealing. However, such a large number of evaluations would be infeasible for a more complex PDE-based model.

Few publications solve the layout problem with gradient-based optimisation algorithms. Lackner and Elkinton (2007) optimised the position of two wind turbines by applying a gradient-based optimisation algorithm to a simplified energy production model with an analytical expression. Huang (2009) combined a genetic algorithm with steepest ascent to accelerate convergence to an optimal solution. With this additional derivative information, the number of iterations was reduced by approximately an order of magnitude to less than 300 iterations, for a similar power extraction. Finally, Fagerfjäll (2010) showed how mixed integer linear programming techniques can be used to optimise for both the number and position of turbines.

All of these publications use very simplified flow models for which the gradient is either available analytically or can be easily approximated. Furthermore, none of the reviewed papers use gradient-based methods with the adjoint technique to find an optimal layout configuration in a physically realistic model.

While this prior research on wind farm optimisation is relevant, there are key differences between this and tidal turbine arrays. Firstly, the flow in a tidal channel is dominantly driven by the predictable tidal forcing, while wind flow modelling is inherently stochastic and needs to include the temporal uncertainty in the magnitude and direction of the wind forcing. Secondly, the ratio of turbine height to free-surface elevation is significantly different: while the rotor diameter of a wind turbine is small compared to the height of the atmosphere, tidal turbines typically have diameters of around 20 m and are deployed in water depths of approximately 50 m or less. This leads to little undisturbed flow

above the turbine which could contribute to wake recovery and thus potentially increases the length of the wake compared to wind turbines (Bryden et al., 2004; Divett et al., 2013).

The remaining sections are organised as follows. In section 2, we formulate the turbine layout problem as an optimisation problem constrained by the shallow water equations. Section 3 discusses the discretisation and implementation, which is then verified in section 4. In section 5, we demonstrate the capabilities of the proposed approach on four idealised scenarios with 32 turbines. Section 6 presents an application of this approach where the positions of up to 256 turbines are optimised in a geometry motivated by the Inner Sound of the Pentland Firth, Scotland. Finally, we make some concluding remarks in section 7.

2. Problem formulation

In this section we formulate the optimal positioning of turbines in an array as a PDE-constrained optimisation problem in the following abstract form:

$$\begin{aligned} \max_{z,m} \ & J(z,m) \\ \text{subject to } & F(z,m) = 0, \\ & b_l \leq m \leq b_u, \\ & g(m) \leq 0, \end{aligned} \tag{1}$$

where $J(z,m) \in \mathbb{R}$ is the functional of interest, m are the design parameters, $F(z,m)$ is a PDE operator parameterised by m with solution z, b_l and b_u are lower and upper bound constraints for the design parameters, and $g(m)$ enforces additional restrictions on the design parameters.

In this work $z = (u, \eta)$ is the solution (horizontal velocity, free-surface displacement) of the shallow water equations $F(z,m) = 0$, and m contains the positions of the turbines. The bounds b_l and b_u are used to enforce that the turbines remain in a prescribed area (here assumed a rectangle for simplicity), while $g(m)$ is used to enforce a minimum distance between any two turbines (e.g. as a multiple of the turbine diameter).

The optimisation problem (1) can be reduced by using the fact that the constraint equation $F(z,m) = 0$ implicitly maps any choice of m to a unique solution z. Hence, the solution z can be considered as an implicit function of the optimisation parameters: $z \equiv z(m)$. By substituting this operator into the functional of interest, we obtain the reduced optimisation problem:

$$\begin{aligned} \max_{m} \ & J(z(m),m) \\ \text{subject to } & b_l \leq m \leq b_u, \\ & g(m) \leq 0. \end{aligned} \tag{2}$$

While it is possible to solve the optimisation problem in unreduced form (1), we choose to solve the reduced form, as it is usually preferable for time-dependent governing equations (Long et al., 2012).

2.1. The design parameters

For the turbine optimisation problem considered here, the design parameter m is a vector containing the positions of the turbines. The x, y turbine positions of N turbines are encoded in the following form:

$$m = \boxed{x_1 \mid y_1 \mid x_2 \mid y_2 \mid \dots \mid x_N \mid y_N}.$$

In some examples, the friction coefficients K_i of the turbines are also allowed to vary, so the vector m becomes

$$m = \boxed{x_1 \mid y_1 \mid K_1 \mid x_2 \mid y_2 \mid K_2 \mid \dots \mid x_N \mid y_N \mid K_N}.$$

This could be further generalised to account for any number of turbine parameters.

2.2. The PDE constraint

The constraint equation $F(z, m) = 0$ enforces the laws of physics in the optimisation problem (1). We assume that the constraint equation fulfills some properties. Firstly, for every m it must yield a unique solution z: hence, we can write z as an implicit function of m. Secondly, F must be differentiable and it is assumed that the derivative with respect to z is continuously invertible.

Let Ω be the domain of interest and let T be the final simulation time. In this work, the physical laws are modelled by the nonlinear shallow water equations:

$$\begin{aligned} \kappa\frac{\partial u}{\partial t} + u\cdot\nabla u - \nu\nabla^2 u + g\nabla\eta + \frac{c_b + c_t(m)}{H}\|u\|u = 0, \\ \kappa\frac{\partial \eta}{\partial t} + \nabla\cdot(Hu) = 0, \end{aligned} \tag{3}$$

where the unknowns $u : \Omega \times (0, T] \to \mathbb{R}^2$ and $\eta : \Omega \times (0, T] \to \mathbb{R}$ are the depth-averaged velocity and the free-surface displacement, respectively, $H : \Omega \to \mathbb{R}$ is the water depth at rest, $g \in \mathbb{R}$ is the acceleration due to gravity, $\nu \in \mathbb{R}$ is the viscosity coefficient, and $c_b \in \mathbb{R}$ and $c_t(m)$ are the coefficients for the quadratic bottom friction and the turbine parameterisation, respectively. The parameter $\kappa \in \{0, 1\}$ specifies if the stationary ($\kappa = 0$) or the non-stationary problem ($\kappa = 1$) is considered; in the stationary case the time-dependency of the variable definitions above can be neglected.

The boundary conditions are as follows: on the domain inflow boundary $\partial\Omega_{\text{in}}$ a Dirichlet boundary condition is applied to the velocity. On the outflow boundary $\partial\Omega_{\text{out}}$ the free-surface displacement is set to zero. Elsewhere, a no-normal flow boundary condition with a free-slip condition for the tangential components is imposed.

2.3. The turbine parameterisation

A turbine is modelled as increased bottom friction in the concerned area (denoted as c_t in equation (3)) (Bryden et al., 2004). This approach is also used in Divett et al. (2013) where the authors set c_t to a constant value at the

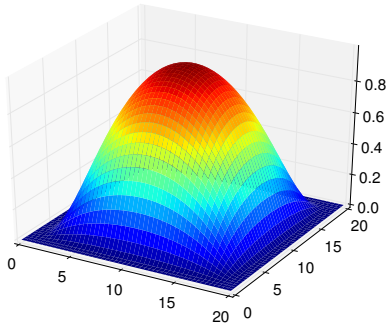


Figure 1: The turbine is parameterised by a smoothly increasing friction coefficient towards the turbine center, given by the bump function (4) multiplied in the x and y-direction.

turbine locations and zero everywhere else. However, this parameterisation is problematic in the context of gradient-based optimisation because the friction becomes a non-differentiable function of the turbine position. For this reason, the turbine parameterisation used here smoothly increases the friction value at each turbine position. The associated friction function is constructed from bump functions, i.e. smooth functions with finite support. A bump function in one dimension is:

$$\psi_{p,r}(x) \equiv \begin{cases} e^{1-1/(1-\|\frac{x-p}{r}\|^2)} & \text{for } \|\frac{x-p}{r}\| < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where the two parameters p and r are the center and the support radius of the bump, respectively. A two-dimensional bump function is obtained by multiplying equation (4) in both independent dimensions. With that, the friction function of a single turbine parameterised by a friction coefficient K_i centred at a point $p_i = (x_i, y_i)$ is given by:

$$C_i(x, y) \equiv K_i \psi_{x_i,r}(x) \psi_{y_i,r}(y). \tag{5}$$

A plot of the resulting friction for $K = 1$, $p = (10, 10)$ and $r = 10$ is shown in figure 1.

The turbine friction function c_t in the governing PDE (3) is defined to be the sum of the friction functions (5) for all N turbines:

$$c_t \equiv \sum_{i=1}^{N} C_i, \tag{6}$$

where a single value for r is used based on the assumption that the deployed turbines are of equal size.

Note that turbine properties can be calibrated by modifying the friction parameter K in equation (5). For example, the amount of energy that is extracted

from an individual turbine (e.g. due to different pitch settings of the turbine blades) can be controlled. It could also be used to handle cut in/out velocities in which the turbines are operational. This extension is not considered in this work. Finally, other more sophisticated turbine parameterisations such as extensions of actuator disc theory could be employed (Roc et al., 2013).

2.4. The functional of interest

The functional of interest J in equation (1) defines the value of interest that is to be maximised. For gradient-based optimisation, J must be differentiable.

A natural choice for the functional of interest is the time-averaged power extracted due to the increased friction in the tidal farm (Vennell, 2012). In the non-stationary case ($\kappa = 1$) this is expressed as:

$$J(u, m) = \frac{1}{2T} \int_0^T \int_\Omega \rho c_t(m) \|u\|^3 \, \mathrm{d}x \, \mathrm{d}t, \tag{7}$$

where ρ is the fluid density. In the stationary case ($\kappa = 0$) the functional is defined to be the power extracted from the increased friction:

$$J(u, m) = \frac{1}{2} \int_\Omega \rho c_t(m) \|u\|^3 \, \mathrm{d}x. \tag{8}$$

Note that this value represents kinetic power extraction rather than electrical power generation, since it does not incorporate losses due to the turbine support structures and the conversion to electricity.

More advanced functional choices could instead maximise the profit of the turbine farm, by including installation and service costs depending on turbine size and the deployment location (Adams et al., 2011). Another alternative would be to incorporate potential environmental impacts. These are not considered in this study.

2.5. Box and inequality constraints

The box and inequality constraints in the generic optimisation problem (1) are used to define the feasible values for the optimisation variables. In the context of the turbine layout problem, a typical condition is to restrict the area in which the turbines may be placed to the development site. The numerical examples in this work have rectangular shaped deployment sites and therefore box constraints are sufficient to enforce this restriction. For more general site shapes appropriate inequality constraints can be used instead.

Another common condition is to ensure that individual turbines do not overlap. This is implemented by enforcing a minimum distance $d_{\min}$ between any two turbines:

$$\|p_i - p_j\|^2 \geq d_{\min}^2 \qquad \forall \; i, j : 1 \leq i < j \leq N. \tag{9}$$

In order for the optimisation to be well-posed, the constraints must satisfy a constraint qualification (Hinze et al., 2009). The box constraints and inequality

constraint (9) are concave functions, and hence satisfy the Concave Constraint Qualification (CCQ) (Carter, 2001, theorem 5.4).

More advanced constraints could for example enforce a minimum or maximum deployment depth or limit the maximum local bathymetry steepness where turbines may be installed, but these are not investigated in this work.

3. Numerical setup

3.1. Optimisation algorithm

A typical gradient-based optimisation algorithm implements the following iteration:

- Choose an initial guess m^0 for the design parameters.
- for $i = 0, 1, ...$
    1. Solve the forward problem $F(z^i, m^i) = 0$ for z^i and evaluate the functional of interest $J(z^i, m^i)$.
    2. Compute the functional gradient $\mathrm{d}J/\mathrm{d}m(z^i, m^i)$.
    3. Find improved design parameters m^{i+1} using the results of 1 and 2.
    4. Stop if the termination criteria are fulfilled.

Optimisation algorithms differ mainly in their implementation of step 3. The main task is to identify an improved parameter choice so that the algorithm converges quickly to the optimal solution while satisfying the imposed constraints.

In this paper we solve the turbine layout problem using sequential quadratic programming (SQP), which is considered to be one of the most efficient optimisation algorithms (Boggs and Tolle, 1995). The implementation used here is the SLSQP algorithm available through the SciPy optimisation package (Jones et al., 2001) and is described in detail in Kraft (1988). The optimisation problem was formulated and solved with the PDE-constrained optimisation framework described in Funke and Farrell (2013).

The SQP implementation used is not scale-invariant, i.e. scaling the functional of interest can impact the convergence of the algorithm. Preliminary numerical investigation found that such rescaling was necessary for achieving fast convergence. For each numerical experiment presented here, the functional of interest was initially scaled such that the maximum value of the initial gradient was ten times the turbine radius.

3.2. The functional gradient computation with the adjoint approach

The second step of the optimisation algorithm requires the computation of the functional derivative with respect to the optimisation parameters $\mathrm{d}J/\mathrm{d}m$. Its efficient computation is achieved using the adjoint approach (Gunzburger, 2003; Giles and Pierce, 2000; Farrell et al., 2012).

The adjoint approach computes the gradient in two steps. Firstly, the adjoint equation is solved to obtain the adjoint solution λ:

$$\frac{\partial F}{\partial z}^{*} \lambda = \frac{\partial J}{\partial z}^{*},$$

where $*$ denotes the Hermitian transpose. Secondly, the functional gradient is obtained by:

$$\frac{\mathrm{d}J}{\mathrm{d}m} = -\lambda^{*}\frac{\partial F}{\partial m} + \frac{\partial J}{\partial m}. \tag{10}$$

For the optimisation problem considered here, the adjoint shallow water equations are:

$$\begin{aligned} -\kappa\frac{\partial \lambda_u}{\partial t} + (\nabla u)^{*}\lambda_u - (\nabla\cdot u)\,\lambda_u - u\cdot\nabla\lambda_u - \nu\nabla^2\lambda_u & \\ -H\nabla\lambda_\eta + \frac{c_b + c_t(m)}{H}\left(\|u\|\lambda_u + \frac{u\cdot\lambda_u}{\|u\|}u\right) = \frac{\partial J}{\partial u}^{*}, & \\ -\kappa\frac{\partial \lambda_\eta}{\partial t} - g\nabla\cdot\lambda_u = 0. & \end{aligned} \tag{11}$$

where $\lambda \equiv (\lambda_u, \lambda_\eta)$ is the vector containing the unknown adjoint velocity and adjoint free-surface displacement, respectively. The derivation of these equations can be found in Funke (2012, appendix C). The non-stationary adjoint equations ($\kappa = 1$) have a final-time condition for the adjoint velocity and free-surface displacement; this condition is homogeneous, as the functional J has no term evaluated at the end of time. The adjoint equations are solved from the final time to the initial time, propagating information backwards in time. The boundary conditions for the adjoint equations are the homogeneous versions of the boundary conditions of the forward equations, again because the functional has no boundary integral terms. The functional derivative $\partial J^{*}/\partial u$ appears as the source term for the adjoint velocity equation and is easy to evaluate as the functional is available as an analytical expression. Note that the adjoint equations are linear while the forward equations are nonlinear, and therefore solving the adjoint equations is typically much cheaper. If the time-dependent equations are solved, the entire forward trajectory is required to assemble the adjoint equations; if the forward trajectory is too large to store at once, then a checkpointing algorithm must be used (Griewank and Walther, 2000; Farrell et al., 2012).

The second step of the adjoint approach (equation (10)) evaluates the functional gradient using the adjoint solution λ. This step only requires the computation of a matrix-vector product and consequently its computational cost is negligible. This allows for the computation of the gradient of the functional with only the solution of one adjoint system, at a cost independent of the number of turbines. This is a key property if many turbines are to be optimised.

In this work, rather than deriving, discretising and implementing the adjoint equation (11) by hand, we apply the high-level algorithmic differentiation approach described in Farrell et al. (2012). This efficiently and automatically

derives and implements the discrete adjoint model from the implementation of the forward model (12), without user intervention. This significantly reduces the effort and expertise required to implement adjoints of complex nonlinear forward models.

3.3. Discretisation

The governing PDEs (3) are discretised with the finite element method. The weak form is derived by multiplying the equations with test functions (Ψ, Φ) from suitable function spaces, integrating over the computational domain and applying integration by parts to selected terms. The weak form of equation (3) is: find (u, η) such that $\forall\ (\Psi, \Phi)$:

$$\begin{aligned} \kappa \left\langle \frac{\partial u}{\partial t}, \Psi \right\rangle_\Omega + \langle u \cdot \nabla u, \Psi \rangle_\Omega + \nu \langle \nabla u, \nabla \Psi \rangle_\Omega \\ + g \langle \nabla \eta, \Psi \rangle_\Omega + \left\langle \frac{c_b + c_t(m)}{H} \|u\| u, \Psi \right\rangle_\Omega &= 0, \\ \kappa \left\langle \frac{\partial \eta}{\partial t}, \Phi \right\rangle_\Omega - \langle Hu, \nabla \Phi \rangle_\Omega + \langle Hu \cdot n, \Phi \rangle_{\partial\Omega_{\text{in}} \cup \partial\Omega_{\text{out}}} &= 0, \end{aligned} \tag{12}$$

where $\langle \cdot, \cdot \rangle_\Omega$ denotes the $L^2(\Omega)$ inner product. The no-normal flow/free-slip conditions on $\partial\Omega \setminus (\partial\Omega_{\text{in}} \cup \partial\Omega_{\text{out}})$ are weakly imposed by excluding the associated surface integrals. The Dirichlet boundary conditions are strongly imposed by restricting the function spaces to functions that yield the correct boundary values.

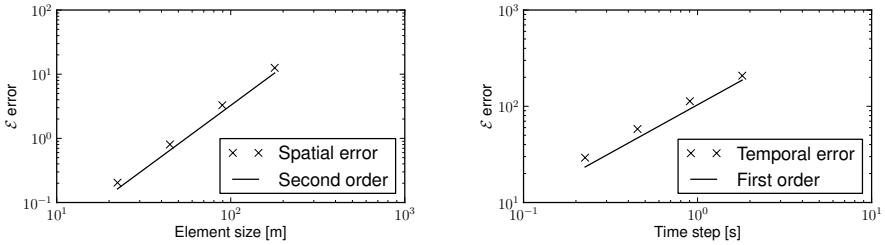
The discretised problem is obtained by choosing discrete function spaces in the weak formulation (12). In this work, these are constructed from a suitable triangulation of the computational domain Ω using the Taylor-Hood finite element pair which uses piecewise quadratic functions on each triangle for the velocity and piecewise linear functions on each triangle for the free-surface displacement (Taylor and Hood, 1973).

If the non-stationary problem is considered ($\kappa = 1$), then the spatially discretised equations (12) must also be discretised in time. In this paper, the time discretisation is performed using the implicit Euler method, due to its unconditional stability and simplicity.

4. Verification

4.1. Verification of the forward model

The shallow water model implementation was verified by order-of-convergence analysis. The analytical solution is constructed using the method of manufactured solutions (Salari and Knupp, 2000; Roache, 2002): a desired analytical solution is chosen and then substituted into the governing PDE, which yields a non-zero remainder. By adding this remainder as a source term to the governing equations, the selected solution becomes an analytical solution to the modified PDE.



(a) The spatial order of convergence (b) The temporal order of convergence

Figure 2: The expected and achieved orders of convergence for the forward model.

For the following tests, the analytical solution consists of sinusoidal functions for both the velocity and the free-surface displacement:

$$\begin{aligned} u_{\text{exact}}(x,y,t) &= \begin{pmatrix} \eta_0 \sqrt{gH^{-1}} \cos\left(kx - \sqrt{gH}kt\right) \\ 0 \end{pmatrix}, \\ \eta_{\text{exact}}(x,y,t) &= \eta_0 \cos\left(kx - \sqrt{gH}kt\right), \end{aligned} \tag{13}$$

with $k = \pi/640$ m^{-1}, $\eta_0 = 2$ m, $H = 50$ m, $\nu = 3$ m^{2}s^{-1}, $c_t = 0$, $c_b = 0.0025$, $g = 9.81$ms^{-2} and a final time of $T = \pi/(\sqrt{gH}k) \approx 28.9$ s which corresponds to half a wave cycle. The computational domain Ω is defined to be a rectangle of size 640 m $\times$ 320 m. Substituting the functions (13) into the shallow water equations (3) yields a non-zero remainder which is added as a source term.

To determine the spatial order of convergence, the time step was fixed to a small value of $\Delta t = T/16800$ s, to ensure that the numerical error of the spatial discretisation dominates the overall discretisation error. Then, the forward model with the added source term was run on four uniform, increasingly fine meshes and the error in the numerical solution (u, η) measured as:

$$\mathcal{E} = \left(||u - u_{\text{exact}}||^2_{\Omega\times(0,T)} + ||\eta - \eta_{\text{exact}}||^2_{\Omega\times(0,T)} \right)^{\frac{1}{2}}.$$

The resulting errors plotted in figure 2a show the second-order convergence that is expected from the Taylor-Hood finite element pair.

For determining the temporal order of convergence, a mesh with 2.5 m element size in the x-direction and 160 m element size in the y-direction was generated (the analytical solution does not vary in the y-direction and hence a relatively large mesh element size can be used). This mesh resolution ensures that the numerical error of the temporal discretisation dominates the overall discretisation error. The forward model was then run with a set of different time steps. The resulting errors plotted in figure 2b show the first-order convergence expected for the implicit Euler time discretisation.

4.2. Verification of the gradient computation

The gradient computation was verified using the Taylor remainder convergence test. Let $\hat{J}(m) \equiv J(z(m), m)$. The first-order Taylor expansion states that:

$$\left| \hat{J}(m + \delta m) - \hat{J}(m) - \frac{d\hat{J}}{dm}\delta m \right| = \mathcal{O}(\|\delta m\|^2). \tag{14}$$

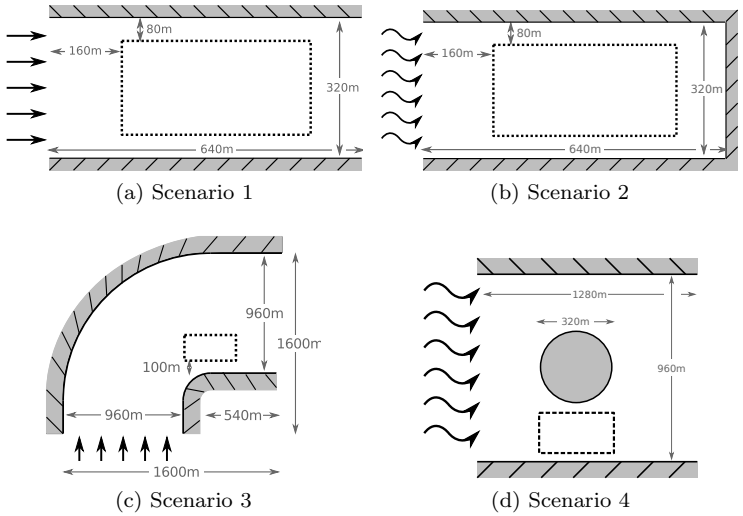
Examining the convergence order of the remainder term is a strong test that the adjoint model and the gradient evaluation are implemented correctly: for nonlinear functionals, the gradient computed using the discrete adjoint approach is correct if and only if the Taylor remainder converges at second order.

Firstly, a simple configuration with a single turbine was set up with the turbine parameterisation and the functional of interest as described above. The Taylor remainder convergence test in many random perturbation directions δm yielded the expected second-order convergence (not shown). Secondly, the Taylor remainder convergence test was applied on several of the numerical examples presented in the following section (not shown). All yielded second-order convergence, giving confidence that the adjoint model and the gradient computation are implemented correctly.

5. Examples

The following numerical examples solve the optimal layout problem in four idealised scenarios motivated by Draper et al. (2010) (figure 3). In all examples, 32 turbines are to be deployed in a rectangular turbine site of size 320 m $\times$ 160 m. The idealised domains simplify the subsequent interpretation of the optimised turbine layouts. Nevertheless, the domains are chosen such that they resemble structures that can be found in practical deployment sites. The main three objectives for the numerical examples are to investigate: by how much can layout optimisation increase the energy extraction? Can the optimisation algorithm reliably improve the energy extraction for different scenarios? How does the choice of the optimisation variables and the constraints impact the resulting layout?

The parameter choices for the experiments are listed in table 1. For the numerical experiments in this paper, the stopping accuracy of the SLSQP optimisation algorithm was set to 10^{-6} unless otherwise stated. This tolerance is extremely tight, as can be seen in the convergence plots, and could be weakened for efficiency reasons. Scenarios 1 and 3 were modelled using the stationary shallow water equations with the following boundary conditions. On the inflow boundary a constant inflow velocity of 2 ms^{-1} was enforced and on the outflow boundary the free-surface displacement was set to zero. A no-normal flow boundary condition with a free-slip condition for the tangential components was applied on the remaining boundaries. Scenarios 2 and 4 solved the non-stationary shallow water equations with boundary conditions explained in the associated sections.



(a) Scenario 1 (b) Scenario 2

(c) Scenario 3 (d) Scenario 4

Figure 3: The four turbine layout scenarios considered in the numerical examples, motivated by Draper et al. (2010). The dashed lines mark the 320 m $\times$ 160 m sized sites where 32 turbines are to be deployed. In scenarios 1 and 3 a constant inflow velocity is enforced, while the other scenarios are driven by a sinusoidal inflow.

All examples used unstructured meshes with a uniform mesh element size of $h = 20$ m outside the site area. Inside the site area, the mesh was structured with an element size of $h = 2$ m. The higher resolution in the turbine site ensures that each individual turbine is well resolved. Doubling the resolution for the problem considered in section 5.1 changed the power extracted by less than 0.5%. It is therefore assumed that the problems are sufficiently well resolved. The resulting meshes consisted of approximately $33,000$ triangles for scenario 1 and 2, $63,000$ triangles for scenario 3 and $45,000$ triangles for scenario 4. All meshes were generated using Gmsh (Geuzaine and Remacle, 2009).

In all numerical experiments, the optimisation algorithm was initialised with the 32 turbines deployed in a regular 8×4 grid and with box constraints for the turbine positions to ensure that the turbines remain inside the site areas.

5.1. A single turbine

As a preliminary test, a single turbine was deployed in the setup of scenario 1. The turbine was placed $640/3$ m $\times 320/2$ m away from the bottom left corner, as shown in figure 4a.

This setup was used to study the dependency of the power extraction J on the friction coefficient K that occurs in the turbine parameterisation (5). Figure 4c shows the power extraction for a range of K values. The graph shows a defined single peak where the power extraction is maximised, and is similar

Parameter	Value
Water depth	$H = 50$ m
Viscosity coefficient	$\nu = 3$ m^{2}s^{-1}
Turbine friction coefficient	$K = 21$
Acceleration due to gravity	$g = 9.81$ ms^{-2}
Water density	$\rho = 1,000$ kgm^{-3}
Bottom friction coefficient	$c_b = 0.0025$
Turbine radii	$r = 10$ m

Table 1: The parameter values used in the experiments of section 5. The non-dimensional bottom friction coefficient is a common value for coastal modelling (Vennell, 2012).

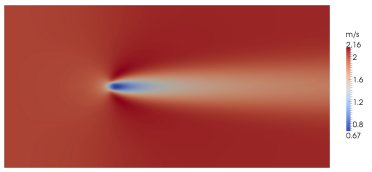
to previous studies (Vennell, 2011, 2012). The reason for this peak is that as $K \to 0$, the turbine friction function c_t approaches 0, which in turn results in no power extraction since the power function (8) is multiplied by c_t; similarly, as $K \to \infty$, the flow is deflected around the turbine and results in the observed power drop. The power extraction peaks for $K = 21$, which was used for all following numerical tests if not otherwise stated. With this choice the single turbine extracted 1.6 MW from the flow.

5.2. Scenario 1

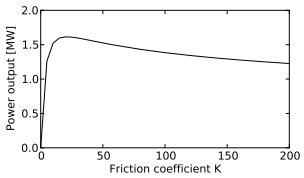
Firstly, the layout problem for scenario 1 (figure 3a) was solved without enforcing a minimum distance between turbines, i.e. the turbines can be placed arbitrarily inside the site area and may even overlap. With that setup the optimisation algorithm terminated after 107 iterations (103 gradient evaluations, 179 functional evaluations). The results are shown in figure 5. Compared to the initial regular layout (figure 5a) the optimisation algorithm was able to increase the farm power extraction by 75% from 27.2 MW to 47.7 MW (figure 5c). The optimised layout (figure 5b) has the turbines aligned in the shape of two $\supset$s with the open end facing the inflow. An intuitive interpretation of this layout is that the water is funneled by the two sides of the $\supset$s and then forced through the dense turbine 'wall' at their closed ends. This interpretation is confirmed by an increasing free-surface displacement (not shown) and velocity difference along the sides of the $\supset$s and the large jump along their closed end (figure 5d).

An additional experiment was performed where inequality constraints were included to enforce a minimum distance of 30 m (3 turbine radii) between each turbine. With this setup, the optimisation algorithm terminated after 76 iterations (75 gradient evaluations, 101 functional evaluations) and the farm power extraction increased by 37% from 27.2 MW to 37.2 MW. The reduced optimised power extraction compared to the previous setup is expected since the inequality constraints add further restrictions to the feasible turbine positioning. In particular, the previous optimised turbine layout is not a feasible solution for this setup.

(a) The turbine friction



(b) The velocity magnitude



(c) The dependency of the power extraction J on the friction coefficient K in equation (5)

Figure 4: The results of deploying a single turbine in the domain of scenario 1.

The optimised alignment differs significantly from the previous one (figure 6b). The two main characteristic structures are a $>$ shaped alignment close to the inflow boundary and a $\mathcal{X}$ shaped turbine wall near the outflow boundary. Furthermore, the turbines are staggered to avoid placing one turbine in the direct wake of another turbine. Finally, the free-surface displacement (not shown) and the velocity magnitude decrease more gradually towards the outflow compared to the previous setup (figure 6d).

5.3. Scenario 2

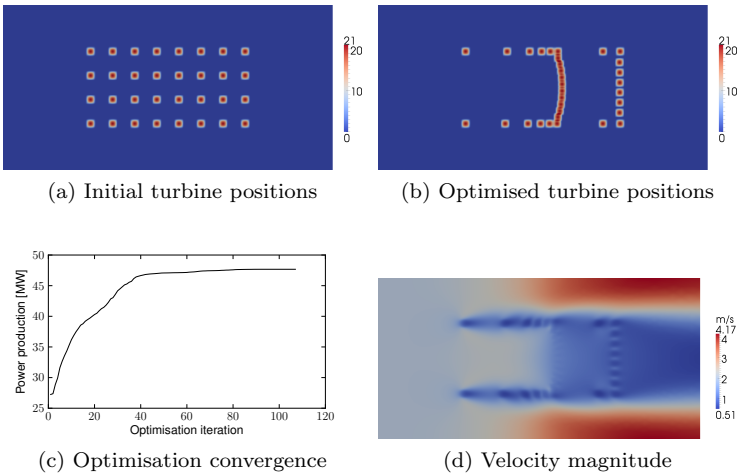
The layout problem for scenario 2 (figure 3b) was solved using the non-stationary shallow water equations.

The boundary conditions were as follows. On the top, bottom and right boundaries, a no-normal flow boundary condition with a free-slip condition for the tangential components was applied. On the left boundary, a Dirichlet boundary condition enforced a sinusoidal in-/outflow velocity:

$$u(x, y, t) = \begin{pmatrix} -2\sin\left(2\pi t/P\right) \\ 0 \end{pmatrix},$$

where P is the tidal period time. Due to the small basin size, a realistically long tidal period would lead to an excessively large tidal range. Therefore, the tidal period was defined to be $P \equiv 10$ minutes, which resulted in a tidal range of ± 12 m. The simulation time was set to one full tidal period with a time step of $\Delta t = 12$ s.

The optimisation was performed without enforcing a minimum distance between the turbines. After 100 optimisation iterations (101 gradient evaluations,



(a) Initial turbine positions (b) Optimised turbine positions

(c) Optimisation convergence (d) Velocity magnitude

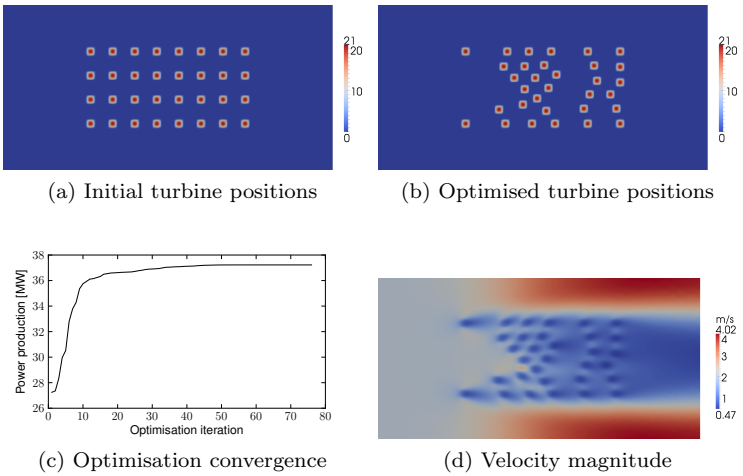
Figure 5: Results of scenario 1 with no inequality constraints, i.e. turbines may overlap.

202 functional evaluations) the relative functional improvement in each iteration dropped below the tolerance and the optimisation was terminated. The results are shown in figure 7. The average power extracted during one tidal cycle increased by 95.4% from 5.24 MW to 10.24 MW (figure 7e). The optimised layout (figure 7b) resembles the result of scenario 1 (figure 5b), with the difference that the openings of the $\sqsupset$s structures face the closed basin side.

5.4. Scenario 3

The domain of the third scenario is shown in figure 3c. For this test, inequality constraints were applied to enforce a minimum distance of 30 m between each turbine. The optimisation algorithm terminated after 36 iterations (35 gradient evaluations, 57 functional evaluations). The optimised farm layout extracts 20.3 MW, which corresponds to an increase of 31% compared to the initial layout (15.5 MW) (figure 8d). The optimised layout features a distinct $\Diamond-$shaped alignment with an opening on the inflow facing side (figure 8a). Figure 8c shows the velocity magnitude and suggest that this hole acts to trap and push the flow through the downstream turbines similar to the previous examples.

Scenario 3 was also used to investigate the potential of additionally optimising the friction coefficients K in the turbine parameterisation (5). For that, the optimisation problem was altered such that both the turbine positions and the K values for each individual turbine were used as optimisation variables. Each K coefficient was constrained such that $0 \leq K \leq 21$. With this setup, the optimisation algorithm terminated after 81 iterations (77 gradient evaluations, 123 functional evaluations). The results are presented in figure 9. Compared to the initial layout, the farm power extraction increased by 38% from 15.5 MW



(a) Initial turbine positions

(b) Optimised turbine positions

(c) Optimisation convergence

(d) Velocity magnitude

Figure 6: Results of scenario 1 with inequality constraints.

to 21.4 MW – the additional freedom of varying the K coefficients resulted in a higher optimised power extraction than the previous setup.

The optimised turbine layout is similar in shape to the previous solution but with a less distinct hole on the inflow facing side. Most notably, the friction coefficients of most turbines are significantly reduced. Only the turbines on the downflow edges of the $\Diamond$ take the maximum value, but nevertheless this configuration extracts 6% more energy than the previous solution where only the positions were optimised. This example shows that optimising the friction coefficient (which can be viewed as reducing the size of the turbine or controlling the blade pitch) can lead to a significant increase in the power extraction of the farm.

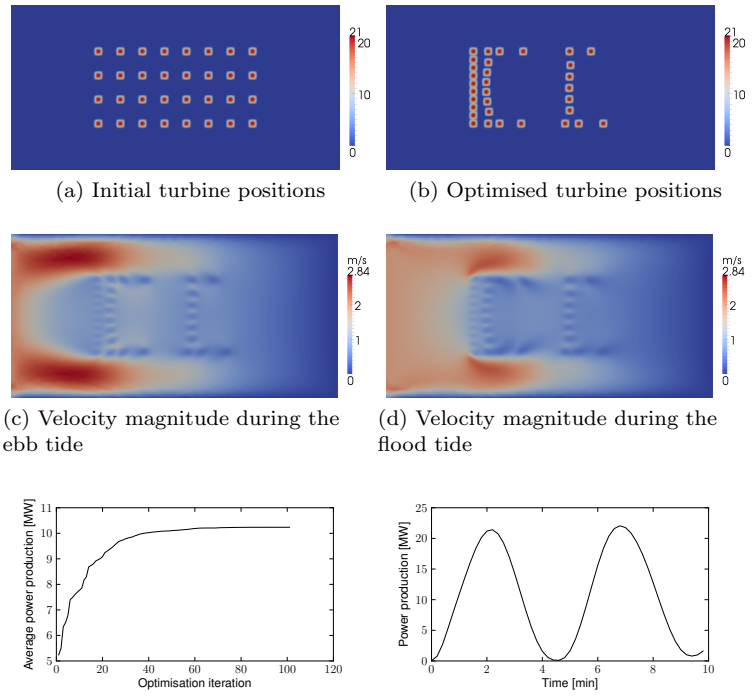
5.5. Scenario 4

For the final scenario (figure 3d), the non-stationary shallow water equations were solved. The simulation time consisted of one 12 h sinusoidal period with a time step of $\Delta t = 864$ s. Dirichlet boundary conditions on the left and right boundaries enforced the following sinusoidal in-/outflow velocity:

$$u(x, y, t) = \begin{pmatrix} -2 \sin(2\pi t/P) \\ 0 \end{pmatrix},$$

with a $P \equiv 12$ h period. On the remaining boundaries, a no-normal flow boundary condition with a free-slip condition for the tangential components was applied.

The setup optimised the turbine positions and applied the inequality constraints to enforce a minimum turbine distance of 30 m. The optimisation



(a) Initial turbine positions

(b) Optimised turbine positions

(c) Velocity magnitude during the ebb tide

(d) Velocity magnitude during the flood tide

(e) Optimisation convergence

(f) Power extraction over time of the optimised configuration

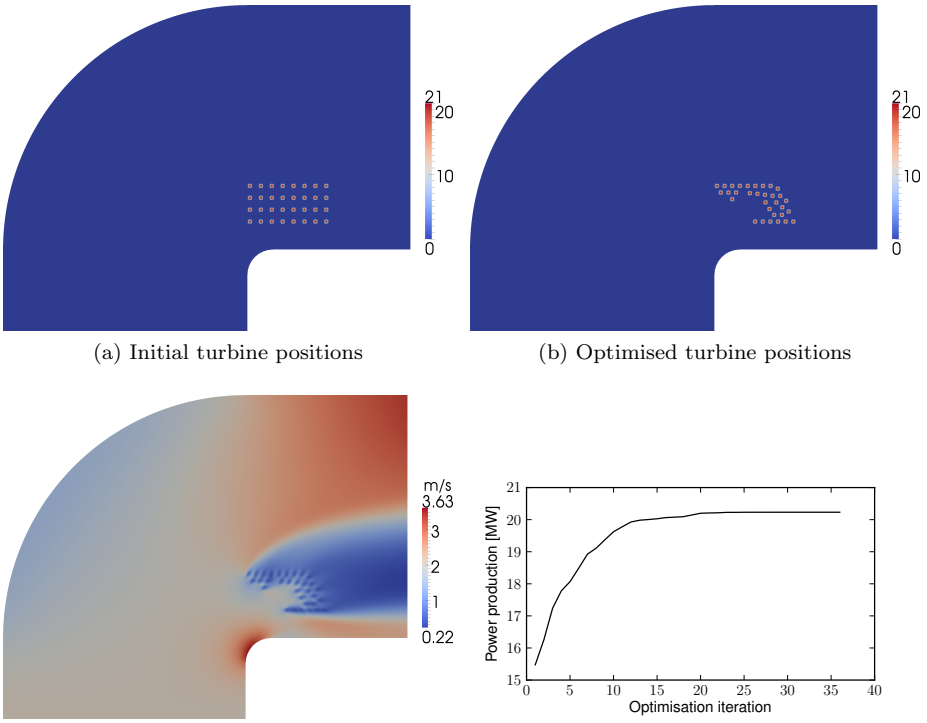
Figure 7: Results of scenario 2.

algorithm terminated after 65 iterations (64 gradient evaluations, 94 functional evaluations). The results are shown in figure 10.

The averaged power extracted during one cycle increased by 22% from 24.21 MW to 29.50 MW (figure 10e). Since the computational domain is symmetric and the simulation time covered one full period, the optimised layout is expected to be symmetric in the x-direction. The numerical solution, shown in figure 10a, indeed shows an almost symmetric solution. The turbine alignment consists of two distorted $\vee$ shapes whose open ends face the in/-outflow boundaries. Similar to the previous example, an interpretation of this alignment is to divert the stream towards the corner of the $\vee$ where turbines can extract large amounts of power. An additional row of turbines can be seen parallel to the bottom of the domain. These turbines are positioned to capture energy from the flow passing along the boundary.

6. Farm optimisation in the Inner Sound of the Pentland Firth

A key design feature of the framework is that it should scale to problems in realistic oceanographic domains with large numbers of turbines in parallel.



(a) Initial turbine positions

(b) Optimised turbine positions

(c) Velocity magnitude

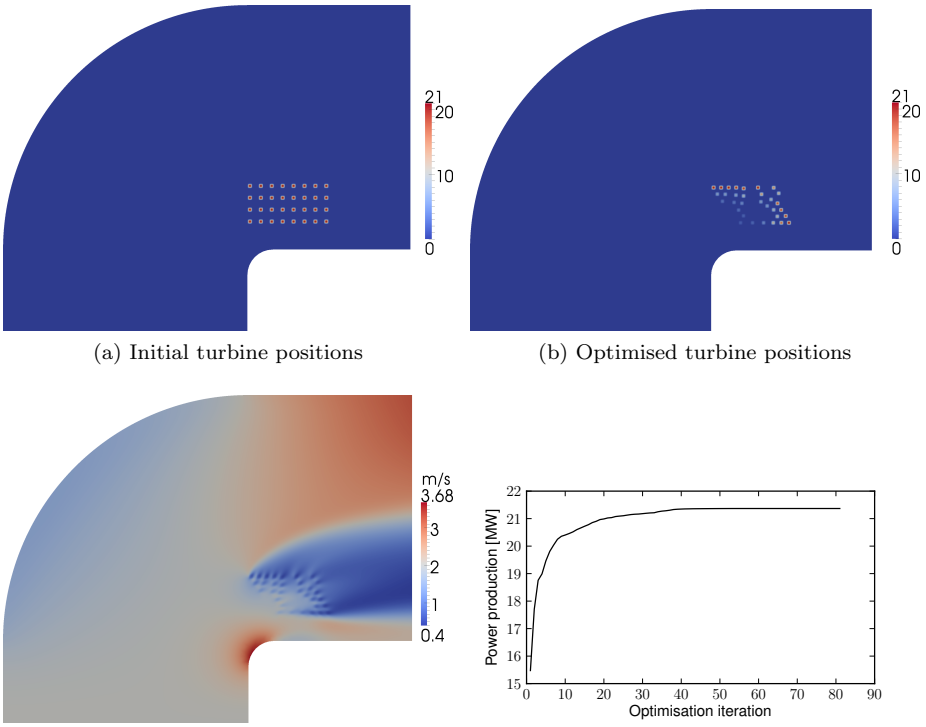
(d) Optimisation convergence

Figure 8: Results of scenario 3 with inequality constraints.

To demonstrate this capability, the optimisation of a farm in a semi-idealised geometry modelled on the Inner Sound of the Pentland Firth (figure 12a) was conducted on the Stampede supercomputer. This site is one of the most promising locations in the UK, and is currently under development by MeyGen Ltd.

The computational domain is shown in figure 12b. The pink area marks the turbine site location which roughly approximates the area used by the MeyGen project. The discretised domain consists of a regular mesh in the turbine site area with 2 m element size, and an unstructured mesh elsewhere with element sizes ranging from $1.5 - 200$ m. The mesh was generated using Gmsh (Geuzaine and Remacle, 2009) using the GSHHS shoreline database (Wessel and Smith, 1996). The mesh consists of 1.25×10^6 elements, which induces a discretisation with a total of 5.6×10^6 degrees of freedom with the Taylor-Hood finite element pair. In this idealised problem, the bathymetry is assumed constant at $H = 50$ m. The addition of bathymetry is straightforward, and will be presented in future work, but would obscure the physical interpretation of the results presented here.

The farm optimisation was modelled during the flood tide using the stationary shallow water equations. The simulations were performed with the same



(a) Initial turbine positions

(b) Optimised turbine positions

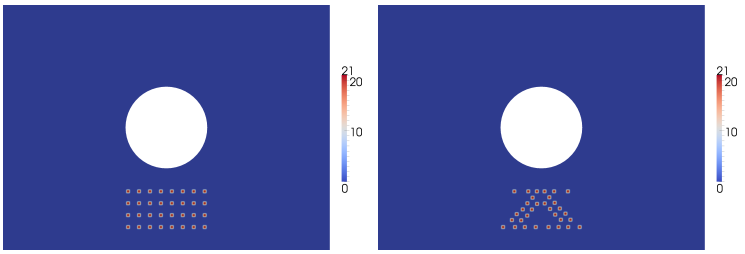
(c) Velocity magnitude

(d) Optimisation convergence

Figure 9: Results of scenario 3 with inequality constraints and optimising for the turbine friction coefficients as well as for the turbine positions.

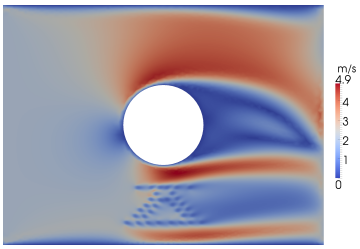
parameter settings as in the previous section (table 1), but with an increased viscosity coefficient of $\nu = 30$ m^{2}s^{-1} and a decreased turbine friction coefficient of $K = 10.5$. This was to ensure the existence of a steady-state solution. Again, the unsteady case is straightforward and will be presented in future work. For efficiency reasons, the convergence tolerance of the SLSQP optimisation algorithm was changed to 1. The same boundary conditions were used as in the steady-state scenarios of the previous section, with the inflow condition imposed on the western boundary and the outflow condition enforced on the eastern boundary.

Two optimisation runs were conducted, with 128 and 256 turbines respectively. Both were performed on the Stampede supercomputer at the Texas Advanced Computing Center on 64 cores; both took between 24 and 48 hours of real time to complete (one functional evaluation took approximately 270 seconds, one gradient evaluation took approximately 90 seconds). The 128 turbine case converged in 197 iterations (197 gradient evaluations, 349 functional evaluations), and increased the power extraction from 300 MW to 371 MW, an increase of 24% (figure 11a). The 256 turbine case converged in 133 iterations

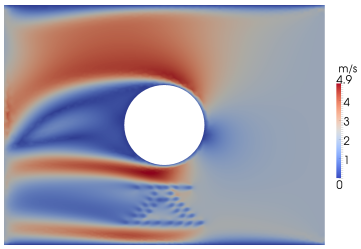


(a) Initial turbine positions

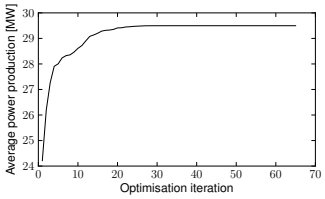
(b) Optimised turbine positions



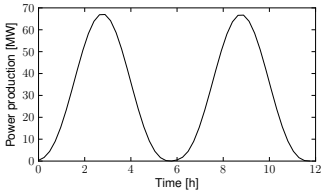
(c) Velocity magnitude at the time when the input velocity from the left reaches its maximum



(d) Velocity magnitude at the time when the input velocity from the right reaches its maximum



(e) Optimisation convergence



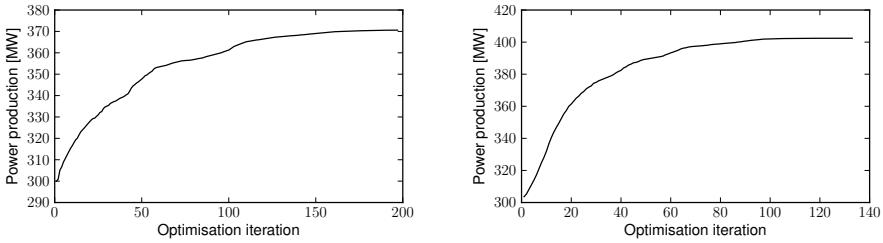
(f) Power extraction over time of the optimised configuration

Figure 10: Results of the non-stationary scenario 4.

(133 gradient evaluations, 185 functional evaluations), and increased the power extraction from 304 MW to 402 MW, an increase of 33% (figure 11b). Even though eight times as many turbines are to be optimised compared to the previous section, the number of iterations has hardly increased at all, suggesting the suitability of the method for larger arrays.

The initial and optimised configurations are shown in figure 13. In both cases, the optimisation algorithm builds very similar structures. The key features of the optimised layouts are the following:

- Walls of turbines aligned with the north and south boundaries; the southern wall extends for the entire zonal length, while the northern wall only extends for the eastern two-thirds of the site. The turbine density on the western side of the northern wall appears to decay in a similar way in both



(a) Optimisation convergence (128 turbines) (b) Optimisation convergence (256 turbines)

Figure 11: Convergence of the Pentland Firth optimisation.

cases.

- Eastern and western 'barrages' of densely packed turbines; in the 128 turbine case, the barrages consist of a single meridional column, while for the 256 turbine case the extra turbines are used to form additional columns. The optimisation algorithm automatically staggers these extra columns to minimize wake shadowing (figure 13d). These barrages are aligned perpendicular to the flow field (figures 14a and 14b) and extract the majority of the power (figures 14c and 14d).

- 'Spurs' arcing from the southern wall in a north-easterly direction; in the 128 turbine case two spurs can be seen, while there are three spurs in the 256 turbine case. Again, the optimisation aligns the spurs perpendicular to the flow field. The spur length is chosen such that the majority of streamlines only intersect with two rows of turbines (figures 14a and 14b).

We hypothesise that these structures serve the following functions:

- The main purpose of the southern and northern walls is to funnel the flow into and retain the flow inside the site boundary. Similar features are found in all scenarios of the previous section. The water predominantly flows in from the northwest, which is why the northern wall stops short on the western side. We hypothesise that the decay of the turbine density on the western side of the northern wall acts to funnel water through the barrages. A similar density decay is evident in the optimised layout of scenario 1 (figure 5b).

- The flow trapped inside the site domain attempts to escape through the northern and southern walls, which causes the arcing of the western and eastern barrages close to the northern and southern boundaries (figures 14a and 14b). Since the prevailing incoming flow is towards the southeast, more turbines are placed on the southern wall to retain the flow. This motivates the deployment of the spurs on the western side of the southern wall.

(a) (b)

Figure 12: (a): Satellite image of Stroma Island and Caithness (Bing Maps, Microsoft). Satellite imagery is only available for the land and nearshore areas. (b): Computational domain with the turbine site marked in pink.

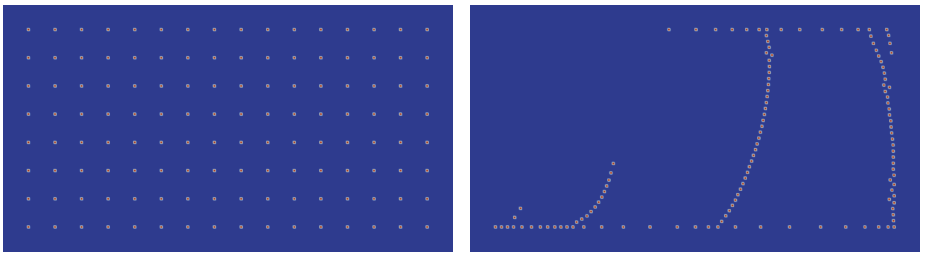
It is not physically meaningful to compare the optimised power extractions for the 128 and 256 turbine cases, as a realistic power curve was not used. The maximum velocity in the site for the 128 turbine case was 3.7 ms^{-1}, while it was 3.0 ms^{-1} for the 256 turbine case. Due to the cubic dependence of the power extraction on the speed, the power per turbine is approximately doubled in the 128 turbine case, and so the total power extraction of the two cases are almost the same. However, if the turbine model enforced a rated speed beyond which no extra power was extracted, this approximate equality would not hold.

7. Conclusions

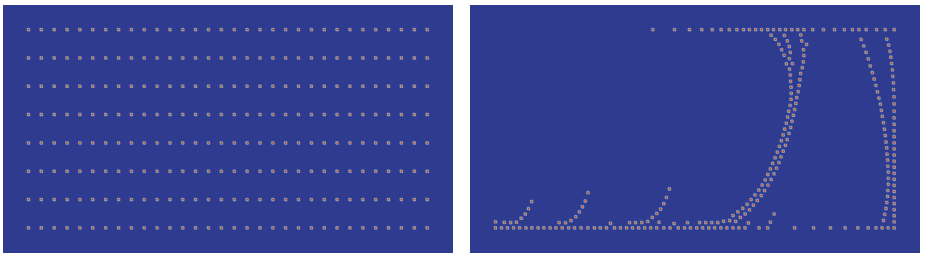
In this work, the optimal layout of tidal turbine farms was formulated as an optimisation problem constrained by partial differential equations. This formulation allows the direct application of sophisticated mathematical optimisation techniques to its solution.

The approach presented here has several key advantages. It fully accounts for the nonlinear interactions between the geometry, the turbines, and the flow throughout the optimisation. The use of gradient-based optimisation algorithms combined with the adjoint technique enables the use of physically-realistic flow models, even for a large number of turbines. Once the flow model inputs are specified (domain, boundary conditions, initial array layout, etc.), the layout optimisation is fully automatic. The approach extends naturally to more realistic flow models, and to different functionals of interest such as profit or environmental impact.

The algorithm was first applied to the optimisation of four idealised scenarios, both to demonstrate the capability of the method and to build physical intuition. In all cases, the optimisation algorithm was successful in significantly increasing the power extracted by the farm, at a computationally feasible cost.



(a) Initial turbine positions (128 turbines) (b) Optimised turbine positions (128 turbines)



(c) Initial turbine positions (256 turbines) (d) Optimised turbine positions (256 turbines)

Figure 13: Initial and optimised turbine positions for the Pentland Firth example.

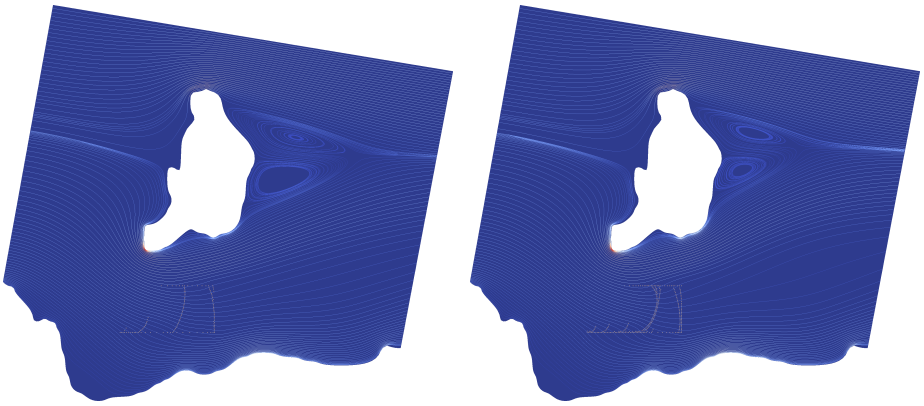
The algorithm was then successfully applied to a more realistic optimisation problem, involving a site of major industrial interest, accurate shoreline geometry, and an industrially relevant number of turbines.

Four main extensions are required to apply this in an industrial setting. Firstly, the transient simulations must be driven by realistic tidal forcing. Secondly, bathymetric effects must be accounted for. Thirdly, the flow model must then be validated against real-world measurements. Finally, the wake modelling should be improved via a turbulence closure and a realistic power curve used. All of these advances are the subject of ongoing work.

The source-code of the turbine layout optimisation program and all examples are open-source and available at `http://opentidalfarm.org`.

8. Acknowledgements

This work is supported by the Grantham Institute for Climate Change, a Fujitsu CASE studentship, EPSRC grants EP/I00405X/1 and EP/J010065/1, an EPSRC Pathways to Impact award, and a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory. High-performance Computing support was provided by the Texas Advanced Computing Center. The authors would like to acknowledge helpful discussions with S. C. Kramer, A. S. Candy, A. Avdis of Imperial College London, and R. Caljouw and S. Crammond of MeyGen Ltd.



(a) Streamline flow visualisation (128 turbines)(b) Streamline flow visualisation (256 turbines)

(c) Turbine power map (128 turbines) (d) Turbine power map (256 turbines)

Figure 14: The streamline and power results for the Pentland Firth example. The power map displays the integrand of the functional J (section 2.4, equation (8)).

References

Adams, N., Ranford, D., Livingston, D., 2011. Modelling and optimisation of tidal arrays. In: Proceedings of the European Wave and Tidal Energy Conference 2011. Southampton, UK.

Bäck, T., 1996. Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, Oxford, UK.

Bilbao, M., Alba, E., 2009. Simulated annealing for optimization of wind farm annual profit. In: 2nd International Symposium on Logistics and Industrial Informatics. pp. 1–5.

Boggs, P. T., Tolle, J. W., 1995. Sequential quadratic programming. Acta Numerica 4 (1), 1–51.

Bryden, I. G., Couch, S. J., 2007. How much energy can be extracted from moving water with a free surface: A question of importance in the field of tidal current energy? Renewable Energy 32 (11), 1961–1966.

Bryden, I. G., Grinsted, T., Melville, G. T., 2004. Assessing the potential of a simple tidal channel to deliver useful energy. Applied Ocean Research 26 (5), 198–204.

Carter, M., 2001. Foundations of Mathematical Economics. MIT Press, Cambridge, MA.

Divett, T., Vennell, R., Stevens, C., 2013. Optimization of multiple turbine arrays in a channel with tidally reversing flow by numerical modelling with adaptive mesh. Philosophical Transactions of the Royal Society A 371 (1985).

Draper, S., Houlsby, G. T., Oldfield, M. L. G., Borthwick, A. G. L., 2010. Modelling tidal energy extraction in a depth-averaged coastal domain. IET Renewable Power Generation 4 (6), 545–554.

Fagerfjäll, P., 2010. Optimizing wind farm layout – more bang for the buck using mixed integer linear programming. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden.

Farrell, P. E., Ham, D. A., Funke, S. W., Rognes, M. E., 2012. Automated derivation of the adjoint of high-level transient finite element programs. SIAM Journal on Scientific Computing. Accepted.

Funke, S. W., 2012. The automation of PDE-constrained optimisation and its applications. Ph.D. thesis, Imperial College London, London, UK.

Funke, S. W., Farrell, P. E., 2013. A framework for automated PDE-constrained optimisation. ACM Transactions on Mathematical Software. Submitted.

Garrett, C., Cummins, P., 2008. Limits to tidal current power. Renewable Energy 33 (11), 2485–2490.

Geuzaine, C., Remacle, J.-F., 2009. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering 79 (11), 1309–1331.

Giles, M. B., Pierce, N. A., 2000. An introduction to the adjoint approach to design. Flow, Turbulence and Combustion 65 (3-4), 393–415.

Griewank, A., Walther, A., 2000. Algorithm 799: revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Transactions on Mathematical Software 26 (1), 19–45.

Gunzburger, M. D., 2003. Perspectives in Flow Control and Optimization. Advances in Design and Control. SIAM, Philadelphia, PA.

Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S., 2009. Optimization with PDE constraints. Vol. 23 of Mathematical Modelling: Theory and Applications. Springer-Verlag, Berlin, Heidelberg, New York.

Huang, H.-S., 2009. Efficient hybrid distributed genetic algorithms for wind turbine positioning in large wind farms. In: IEEE International Symposium on Industrial Electronics 2009. pp. 2196–2201.

Jones, E., Oliphant, T., Peterson, P., et al., 2001. SciPy: Open source scientific tools for Python.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, 1995. Vol. 4. pp. 1942–1948.

Kraft, D., 1988. A software package for sequential quadratic programming. Forschungsbericht / Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt 88 (28).

Laarhoven, P. J. M., Aarts, E. H. L. (Eds.), 1987. Simulated annealing: theory and applications. Kluwer Academic Publishers, Norwell, MA.

Lackner, M. A., Elkinton, C. N., 2007. An analytical framework for offshore wind farm layout optimization. Wind Engineering 31 (1), 17–31.

Lee, S. H., Lee, S. H., Jang, K., Lee, J., Hur, N., 2010. A numerical study for the optimal arrangement of ocean current turbine generators in the ocean current power parks. Current Applied Physics 10 (2), S137 – S141.

Long, K., Boggs, P. T., van Bloemen Waanders, B. G., 2012. Sundance: high-level software for PDE-constrained optimization. Scientific Programming 20 (3), 293–310.

Naumann, U., 2012. The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation. Vol. 24 of Software, Environments, and Tools. SIAM, Philadelphia, PA.

Roache, P. J., 2002. Code verification by the method of manufactured solutions. Journal of Fluids Engineering 124 (1), 4–10.

Roc, T., Conley, D. C., Greaves, D., 2013. Methodology for tidal turbine representation in ocean circulation model. Renewable Energy 51 (0), 448–464.

Rudolph, G., 1996. Convergence of evolutionary algorithms in general search spaces. In: Proceedings of IEEE International Conference on Evolutionary Computation, 1996. Nagoya, Japan.

Salari, K., Knupp, P., 2000. Code verification by the method of manufactured solutions. Tech. Rep. SAND2000-1444, Sandia National Laboratories, Albuquerque, NM.

Salcedo-Sanz, S., Saavedra-Moreno, B., Paniagua-Tineo, A., Prieto, L., Portilla-Figueras, A., 2011. A review of recent evolutionary computation-based techniques in wind turbines layout optimization problems. Central European Journal of Computer Science 1, 101–107.

Taylor, C., Hood, P., 1973. A numerical solution of the Navier-Stokes equations using the finite element technique. Computers & Fluids 1 (1), 73–100.

Thomson, M. D., Whelan, J. I., Gill, L., 2011. The development of a tool for the design and optimisation of tidal stream turbine arrays. In: Proceedings of the European Wave and Tidal Energy Conference, 2011. Southampton, UK.

Vennell, R., 2010. Tuning turbines in a tidal channel. Journal of Fluid Mechanics 663, 253–267.

Vennell, R., 2011. Tuning tidal turbines in-concert to maximise farm efficiency. Journal of Fluid Mechanics 671 (1), 587–604.

Vennell, R., 2012. The energetics of large tidal turbine arrays. Renewable Energy 48 (0), 210–219.

Wan, C., Wang, J., Yang, G., Zhang, X., 2010. Optimal micro-siting of wind farms by particle swarm optimization. In: Tan, Y., Shi, Y., Tan, K. (Eds.), Advances in Swarm Intelligence. Vol. 6145 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, New-York, pp. 198–205.

Wessel, P., Smith, W. H. F., 1996. A global, self-consistent, hierarchical, high-resolution shoreline database. Journal of Geophysical Research: Solid Earth 101 (B4), 8741–8743.